

An Overview of Trilinos



Jonathan Hu
Sandia National Laboratories

Eleventh DOE ACTS Collection Workshop
August 19th, 2010



Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.





Main Title 32pt

Subtitle 28 pt

Date /time 20pt

Speaker 24pt

Speaker title 22pt

The acknowledgement statement **MUST** be used on the title slide of all presentation material distributed outside of Sandia.



Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.





Outline of Talk

- Background / Motivation.
- Trilinos Package Concepts.
- Whirlwind Tour of Trilinos Packages.
- Getting Started.
- Concluding remarks.
- Hands On Tutorial from 3:45-5:45pm this afternoon



Who are the developers?

(a very incomplete list)

4

Chris Baker

Develops: Anasazi, RBGen, Tpetra

Ross Bartlett

Leads: Thyra and Stratimikos
Develops: Rythmos

Pavel Bochev

Leads: Intrepid

Paul Boggs

Develops: Thyra

Eric Boman

Leads: Isorropia
Develops: Zoltan

Cedric Chevalier

Develops: Zoltan, Isorropia

Todd Coffey

Leads: Rythmos

Eric Cyr

Leads: Teko

David Day

Develops: Komplex and Intrepid

Karen Devine

Leads: Zoltan
Develops: Isorropia

Clark Dohrmann

Develops: CLAPS

Carter Edwards

Leads: ThreadPool, Shards, PhdMesh
Develops: STK

Michael Gee

Develops: Moertel, ML

Glen Hansen

Leads: Moertel

Dave Hensinger

Leads: PamGen

Bob Heaphy

Leads: Trilinos SQA

Mike Heroux

Trilinos Project Leader
Leads: Epetra, AztecOO, Kokkos, Komplex,
IFPACK, Thyra, Tpetra
Develops: Amesos, Belos, EpetraExt, Jpetra

Ulrich Hetmaniuk

Develops: Anasazi

Robert Hoekstra

Leads: EpetraExt
Develops: Epetra, Thyra, Tpetra

Mark Hoemmen

Develops: Anasazi

Russell Hooper

Develops: NOX

Vicki Howle

Leads: Meros
Develops: Belos and Thyra

Jonathan Hu

Develops: ML

Sarah Knepper

Develops: Komplex

Joe Kotulski

Leads: Pliris

Robert Kirby

Develops: Intrepid

Pat Knupp

Develops: Mesquite

Jason Kraftcheck

Develops: Mesquite

Rich Lehoucq

Develops: Anasazi and Belos

Nicole Lemaster

Leads: CTrilinos, ForTrilinos

Kevin Long

Leads: Thyra, Sundance
Develops: Teuchos

Karla Morris

Develops: ForTrilinos, Ctrilinos, Morfeus

Kurtis Nusbaum

Leads: Optika

Ron Oldfield

Leads: Trios

Roger Pawlowski

Leads: NOX, Phalanx
Develops: Shards, LOCA

Brent Perschbacher

Framework manager

Kara Peterson

Develops: Intrepid

Eric Phipps

Leads: Sacado, Stokhos
Develops: LOCA, NOX

Siva Rajamanickam

Develops: Zoltan

Denis Ridzal

Leads: Aristos and Intrepid

Lee Ann Riesen

Develops: Zoltan, Isorropia

Damian Rouson

Leads: Morfeus
Develops: ForTrilinos and Ctrilinos

Chris Siefert

Develops: ML, Ifpack

Greg Sjaardema

Develops: Trios, STK

Andrew Salinger

Leads: LOCA
Develops: Trikota

Bill Spotz

Leads: PyTrilinos
Develops: Epetra, New_Package

Heidi Thornquist

Leads: Anasazi, Belos, RBGen, and Teuchos

Ray Tuminaro

Leads: ML and Meros

Jim Willenbring

Develops: Epetra and New_Package.
Trilinos library manager

Alan Williams

Leads: Isorropia
Develops: Epetra, EpetraExt, AztecOO, Tpetra
Tifpack

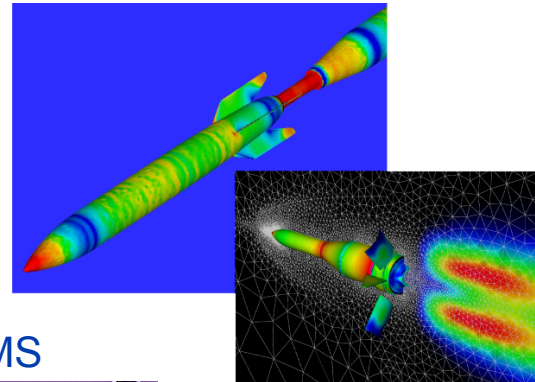
Michael Wolf

Develops: Zoltan, Isorropia

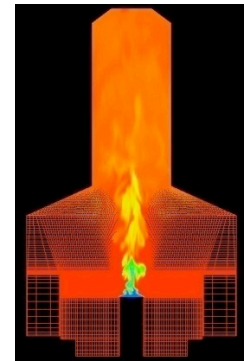
Sandia Physics Simulation Codes

- Element-based
 - ◆ Finite element, finite volume, finite difference, network, etc...
- Large-scale
 - ◆ Billions of unknowns
- Parallel
 - ◆ MPI-based SPMD
 - ◆ Distributed memory
- C++
 - ◆ Object oriented
 - ◆ Some coupling to legacy Fortran libraries

Fluids



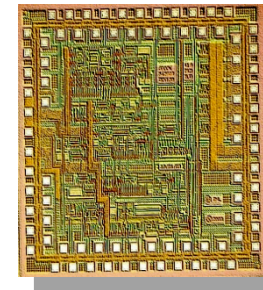
Combustion



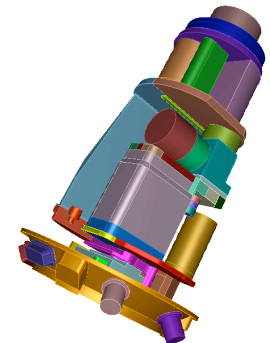
MEMS



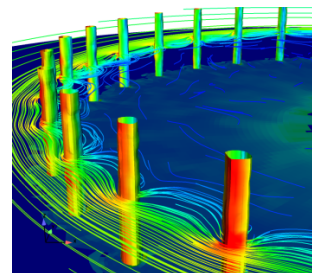
Circuits



Structures



Plasmas





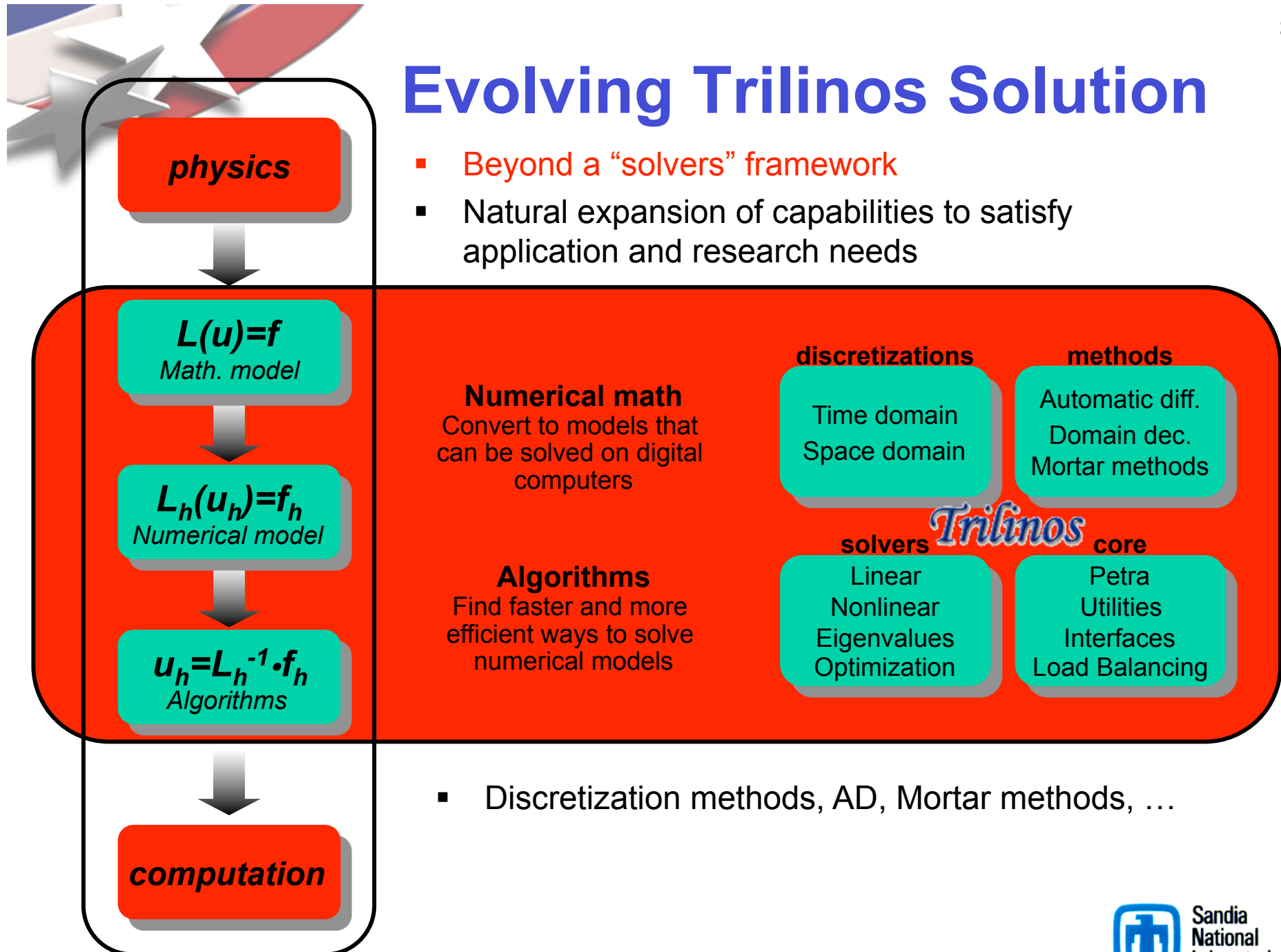
Motivation For Trilinos

- Sandia does LOTS of solver work.
- 10 years ago ...
 - ◆ Aztec was a mature package. Used in many codes.
 - ◆ FETI, PETSc, DSCPack, Spooles, ARPACK, DASPK, and many other codes were (and are) in use.
 - ◆ New projects were underway or planned in multi-level preconditioners, eigensolvers, non-linear solvers, etc...
- The challenges:
 - ◆ Little or no coordination was in place to:
 - Efficiently reuse existing solver technology.
 - Leverage new development across various projects.
 - Support solver software processes.
 - Provide consistent solver APIs for applications.
 - ◆ ASCI was forming software quality assurance/engineering (SQA/SQE) requirements:
 - Daunting requirements for any single solver effort to address alone.

Trilinos – A High Level View

- Trilinos is a framework
 - ◆ Public release 10.4 contains 47 numerical libraries, called “**packages**”
 - ◆ Packages
 - Are developed by domain experts
 - Are often interoperable
 - May have originated within Trilinos, or started separately (e.g., Zoltan)
- Why it's good to be a package
 - ◆ Software version control (git), unified build system (cmake)
 - ◆ Automated nightly and integrated app testing
 - ◆ Mailing lists (dev, user, regression, announce, checkin)
 - ◆ Autonomy (R&D, coding style, testing, documentation, etc.)
- What capabilities does Trilinos provide for my app?
 - ◆ Basic parallel linear algebra
 - ◆ Solvers, AD, meshing, discretizations, optimization, load-balancing, etc.
 - Capabilities all developed and supported by domain experts
 - ◆ Unified abstract API
 - ◆ Lots more on capabilities to come ...

Evolving Trilinos Solution



Trilinos Package Summary

| | Objective | Package(s) |
|------------------------|-----------------------------------|---|
| Discretizations | Meshing & Spatial Discretizations | Intrepid, Phalanx, Shards, Pamgen, Sundance, Mesquite, STK, Moertel |
| | Time Integration | Rythmos |
| Optimization | Optimization (SAND) | MOOCHO, Aristos |
| Methods | Automatic Differentiation | Sacado |
| Core | Linear algebra objects | Epetra, Jpetra, Tpetra |
| | Abstract interfaces | Thyra, Stratimikos, RTOp, Piro |
| | Load Balancing | Zoltan, Isorropia |
| | “Skins” | PyTrilinos, WebTrilinos, Star-P, ForTrilinos, CTrilinos |
| | C++ utilities, (some) I/O | Teuchos, EpetraExt, Kokkos, Triutils |
| | GUIs | Optika |
| Preconditioners | Multigrid methods | ML |
| | Domain decomposition methods | CLAPS, IFPACK |
| | ILU-type methods | AztecOO, IFPACK, TIFPACK |
| | Block preconditioners | Teko, Meros |
| Solvers | Iterative (Krylov) linear solvers | AztecOO, Belos, Komplex |
| | Direct sparse linear solvers | Amesos |
| | Direct dense linear solvers | Epetra, Teuchos, Pliris |
| | Nonlinear system solvers | NOX, LOCA |
| | Iterative eigenvalue solvers | Anasazi |
| | Stochastic PDEs | Stokhos |



Trilinos Strategic Goals

- **Scalable Computations:** As problem size and processor counts increase, the cost of the computation will remain nearly fixed.
- **Hardened Computations:** Never fail unless problem essentially intractable, in which case we diagnose and inform the user why the problem fails and provide a reliable measure of error.
- **Full Vertical Coverage:** Provide leading edge enabling technologies through the entire technical application software stack: from problem construction, solution, analysis and optimization.
- **Grand Universal Interoperability:** All Trilinos **packages** will be interoperable, so that any combination of solver packages that makes sense algorithmically will be **possible** within Trilinos.
- **Universal Accessibility:** All Trilinos capabilities will be available to users of major computing environments: C++, Fortran, Python and the Web, and from the desktop to the latest scalable systems.
- **Universal Solver RAS:** Trilinos will be:
 - **Reliable:** Leading edge hardened, scalable solutions for each of these applications
 - **Available:** Integrated into every major application at Sandia
 - **Serviceable:** Easy to maintain and upgrade within the application environment.

Algorithmic
Goals

Software
Goals

Target Platforms: Any and All (Now and in the Future)



- Desktop: Development and more...

- Capability machines:

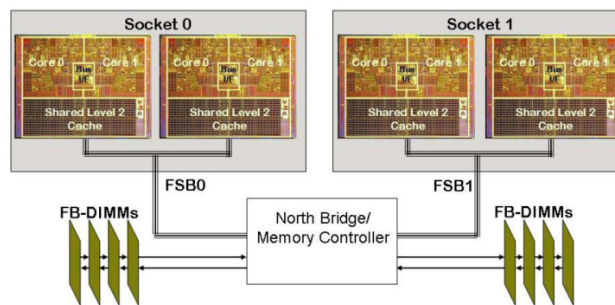
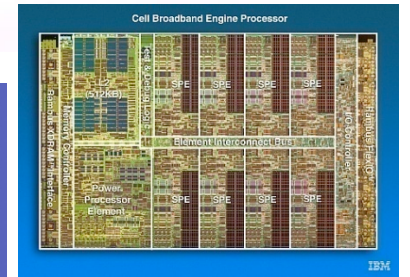
- ◆ Redstorm (XT3), Clusters
- ◆ Roadrunner (Cell-based).
- ◆ Multicore nodes.

- Parallel software environments:

- ◆ MPI of course.
- ◆ UPC, CAF, threads, vectors,...
- ◆ Combinations of the above.

- User “skins”:

- ◆ C++/C, Python
- ◆ Fortran.
- ◆ Web, CCA.





Whirlwind Tour of Packages

Frequently Requested Discretizations Methods Core Solvers/
Preconditioners



Teuchos

- Portable utility package of commonly useful tools:
 - ♦ ParameterList class: key/value pair database, recursive capabilities.
 - ♦ LAPACK, BLAS wrappers (templated on ordinal and scalar type).
 - ♦ Dense matrix and vector classes (compatible with BLAS/LAPACK).
 - ♦ FLOP counters, timers.
 - ♦ Ordinal, Scalar Traits support: Definition of 'zero', 'one', etc.
 - ♦ Reference counted pointers / arrays, and more...
- Takes advantage of advanced features of C++:
 - ♦ Templates
 - ♦ Standard Template Library (STL)
- Teuchos::ParameterList:
 - ♦ Allows easy control of solver parameters.
 - ♦ XML format input/output.

POCs: Roscoe Barlett, Mike Heroux {rabartl,maherou}@sandia.gov



Trilinos Common Language: Petra

- Petra provides a “common language” for distributed linear algebra objects (operator, matrix, vector)
- Petra¹ provides distributed matrix and vector services.
- Exists in basic form as an object model:
 - ◆ Describes basic user and support classes in UML, independent of language/implementation.
 - ◆ Describes objects and relationships to build and use matrices, vectors and graphs.
 - ◆ Has 3 implementations under development.

¹Petra is Greek for “foundation”.



Epetra – Distributed Linear Algebra

- Epetra (Essential Petra):
 - ◆ Current production version.
 - ◆ Restricted to real, double precision arithmetic.
 - ◆ Uses stable core subset of C++ (circa 2000).
 - ◆ Interfaces accessible to C and Fortran users.
 - ◆ Global index size limited to 32-bit integers
- Many sparse matrix formats: CSR, FE-CSR, MSR, VBR, JD
- Common abstract matrix interface: Epetra_RowMatrix
- Vector, MultiVector, Graphs
- Serial & MPI support
- Maps - parallel data management



POCs: Mike Heroux, Alan Williams {maherou,william}@sandia.gov



EpetraExt: Extensions to Epetra

- Library of useful classes not needed by everyone
- Most classes are types of “transforms”.
- Examples:
 - ♦ Graph/matrix view extraction.
 - ♦ Epetra/Zoltan interface.
 - ♦ Explicit sparse transpose.
 - ♦ Singleton removal filter, static condensation filter.
 - ♦ Overlapped graph constructor, graph colorings.
 - ♦ Permutations.
 - ♦ Sparse matrix-matrix multiply.
 - ♦ Matlab, MatrixMarket I/O functions.
 - ♦ Wrapper for PETSc *aij* matrices.
- Most classes are small, useful, but non-trivial to write.

Developers: Robert Hoekstra, Alan Williams, Mike Heroux, many others



Tpetra – Distributed Linear Algebra (Templated)

- Tpetra (Templated Petra):
 - ◆ C++, successor to Epetra.
 - ◆ Templated scalar and ordinal fields.
 - ◆ Uses namespaces, and STL: Improved usability/efficiency.
- Similarities to Epetra
 - ◆ Sparse parallel matrices: CSR, VBR
 - ◆ MultiVector
 - ◆ Graphs
- Important differences
 - ◆ Templated scalar/ordinal fields
 - ◆ Handles only intra-node communication.
 - Inter-node communication is handled by Kokkos package
 - Multicore, GPU
 - ◆ Allows for mixed MPI/threading, MPI/GPU



POCs: Chris Baker bakercg@ornl.gov
Mike Heroux, Alan Williams {maherou,william}@sandia.gov



Belos

- Iterative linear solver library, written in templated C++.
- Provide a generic framework for developing iterative algorithms for solving large-scale, linear problems.
- Algorithm implementation is accomplished through the use of traits classes and abstract base classes:
 - ♦ Operator-vector products: `Belos::MultiVecTraits`, `Belos::OperatorTraits`
 - ♦ Orthogonalization: `Belos::OrthoManager`, `Belos::MatOrthoManager`
 - ♦ Status tests: `Belos::StatusTest`, `Belos::StatusTestResNorm`
 - ♦ Iteration kernels: `Belos::Iteration`
 - ♦ Linear solver managers: `Belos::SolverManager`
- Krylov methods for single systems $Ax=b$
 - ♦ Simultaneously solved systems w/ multiple-RHS: $AX = B$
 - ♦ Sequentially solved systems w/ multiple-RHS: $AX_i = B_i, i=1,...,t$
 - ♦ Sequences of multiple-RHS systems: $A_i X_i = B_i, i=1,...,t$
- Many advanced methods for these types of linear systems
 - ♦ Block methods: block GMRES [Vital], block CG/BICG [O'Leary]
 - ♦ "Seed" solvers: hybrid GMRES [Nachtigal, et al.]
 - ♦ Recycling solvers: recycled Krylov methods [Parks, et al.]
 - ♦ Restarting techniques, orthogonalization techniques, ...

POCs: Heidi Thornquist, Mike Parks, Rich Lehoucq
{hkthorn,mlparks,rblehou}@sandia.gov



IFPACK: Algebraic Preconditioners

- Overlapping Schwarz preconditioners with incomplete factorizations, block relaxations, block direct solves.
- Accept user matrix via abstract matrix interface (Epetra versions).
- Uses Epetra for basic matrix/vector calculations.
- Supports simple perturbation stabilizations and condition estimation.
- Separates graph construction from factorization, improves performance substantially.
- Compatible with AztecOO, ML, Amesos. Can be used by NOX and ML.

POCs: Mike Heroux, Alan Williams



Amesos

- Interface to direct solvers for distributed sparse linear systems (KLU, UMFPACK, SuperLU, MUMPS, ScaLAPACK)
- Challenges:
 - ♦ No single solver dominates
 - ♦ Different interfaces and data formats, serial and parallel
 - ♦ Interface often changes between revisions
- Amesos offers:
 - ♦ A single, clear, consistent interface, to various packages
 - ♦ Common look-and-feel for all classes
 - ♦ Separation from specific solver details
 - ♦ Use serial and distributed solvers; Amesos takes care of data redistribution
 - ♦ Native solvers: KLU and Paraklete

POCs: Mike Heroux maherou@sandia.gov



: Multigrid Preconditioners

- Smoothed aggregation multigrid, domain decomposition preconditioning, nonsymm. multigrid
- Critical technology for scalable performance of some key apps.
- **ML compatible with other Trilinos packages:**
 - ◆ Provides internal smoothers + smoothers from Ifpack, Amesos, AztecOO
 - ◆ If user data can be wrapped as Epetra_RowMatrix object (abstract interface), ML can be applied to it.
 - ◆ ML preconditioners can precondition methods in AztecOO, Belos, Anasazi.
- **Coarsening strategies:** uncoupled, MIS, 3rd party graph
- **Smoothers:** Jacobi, SOR, Chebyshev, ILU/IC/etc.
- **Problem types:** Laplace, electro-magnetics, convection-diffusion, elasticity
- Can also be used in stand-alone mode.

POCs: Ray Tuminaro, Jonathan Hu, Chris Siefert {rstumin,jhu,csiefer}@sandia.gov



Anasazi

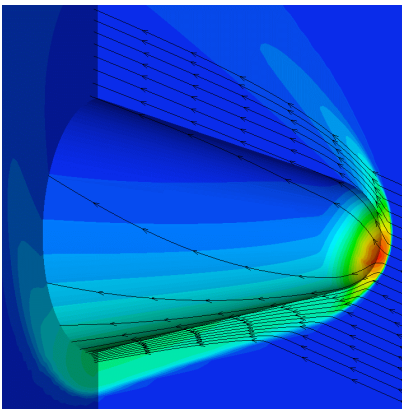
- Eigensolver library, written in templated C++.
- Provides a generic framework for developing iterative algorithms for solving large-scale eigenproblems.
- Algorithm implementation is accomplished through the use of traits classes and abstract base classes:
 - ♦ Operator-vector products: `Anasazi::MultiVecTraits`, `Anasazi::OperatorTraits`
 - ♦ Orthogonalization: `Anasazi::OrthoManager`, `Anasazi::MatOrthoManager`
 - ♦ Status tests: `Anasazi::StatusTest`, `Anasazi::StatusTestResNorm`
 - ♦ Iteration kernels: `Anasazi::EigenSolver`
 - ♦ Eigensolver managers: `Anasazi::SolverManager`
 - ♦ Eigenproblem: `Anasazi::Eigenproblem`
 - ♦ Sort managers: `Anasazi::SortManager`
- Currently has solver managers for three eigensolvers:
 - ♦ Block Krylov-Schur
 - ♦ Block Davidson
 - ♦ LOBPCG
- Can solve:
 - ♦ standard and generalized eigenproblems
 - ♦ Hermitian and non-Hermitian eigenproblems
 - ♦ real or complex-valued eigenproblems

POCs: Heidi Thornquist, Rich Lehoucq, Chris Baker
{hkthorn,rblehou}@sandia.gov, bakercg@ornl.gov

NOX: Nonlinear Solvers

- Suite of nonlinear solution methods

Broyden's Method



Newton's Method



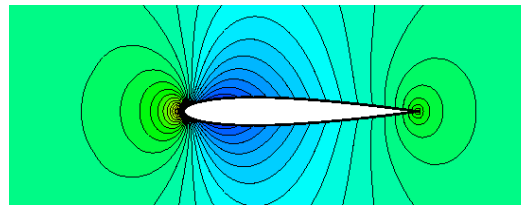
Globalizations

Line Search

Interval Halving
Quadratic
Cubic
More'-Thuente

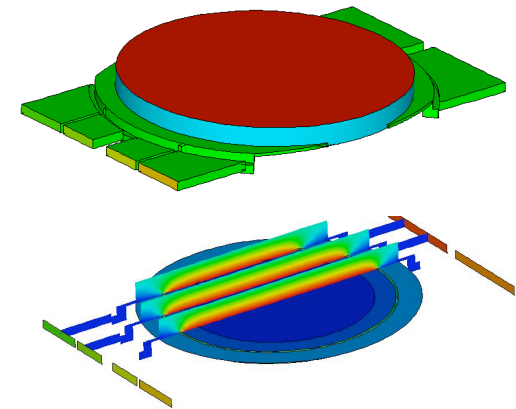
Trust Region

Dogleg
Inexact Dogleg



<http://trilinos.sandia.gov/packages/nox>

Tensor Method



Jacobian Estimation

- Graph Coloring
- Finite Difference
- Jacobian-Free Newton-Krylov

Implementation

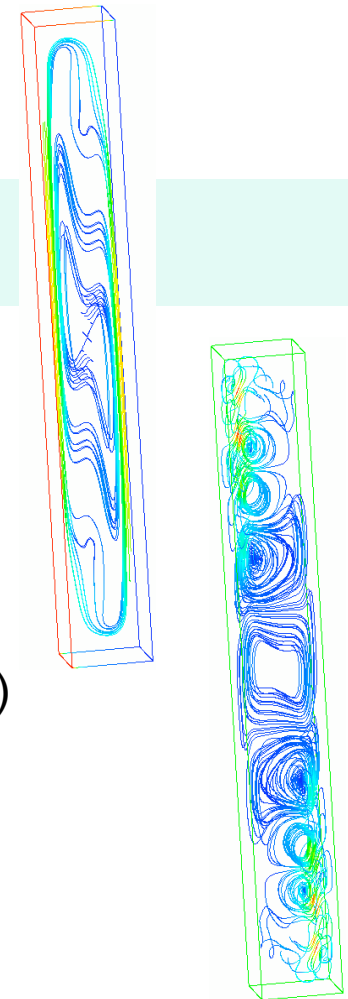
- Parallel
- OO-C++
- Independent of the linear algebra package!

POCs: Roger Pawlowski, Eric Phipps, Andy Salinger
{rppawlo,etphipp,agsalin}@sandia.gov



LOCA

- Library of continuation algorithms
- Provides
 - ◆ Zero order continuation
 - ◆ First order continuation
 - ◆ Arc length continuation
 - ◆ Multi-parameter continuation (via Henderson's MF Library)
 - ◆ Turning point continuation
 - ◆ Pitchfork bifurcation continuation
 - ◆ Hopf bifurcation continuation
 - ◆ Phase transition continuation
 - ◆ Eigenvalue approximation (via ARPACK or Anasazi)



POCs: Andy Salinger, Eric Phipps
{agsalin,etphipp}@sandia.gov



Whirlwind Tour of Packages

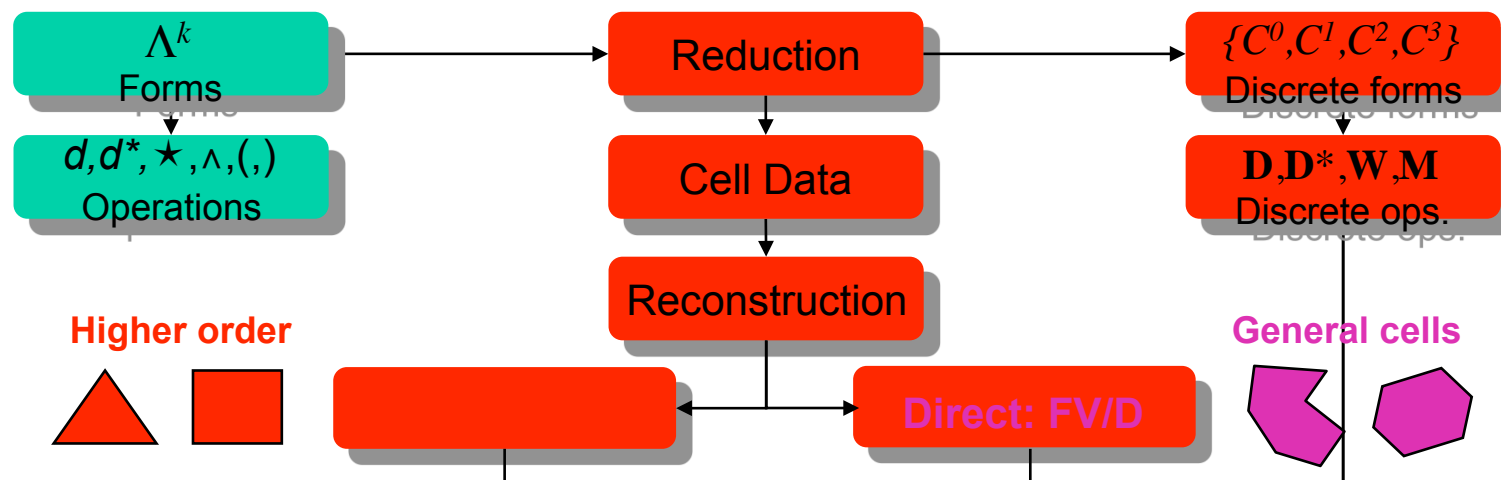
Frequently Requested **Discretizations** Methods Core Solvers/
Preconditioners

Intrepid

*Interoperable Tools for Rapid Development
of Compatible Discretizations*

Intrepid offers an **innovative software design** for compatible discretizations:

- allows access to FEM, FV and FD methods using a common API
- supports **hybrid discretizations** (FEM, FV and FD) on unstructured grids
- supports a variety of cell shapes:
 - standard shapes (e.g. tets, hexes): high-order finite element methods
 - arbitrary (polyhedral) shapes: low-order mimetic finite difference methods
- enables optimization, error estimation, V&V, and UQ using fast invasive techniques (direct support for cell-based derivative computations or via automatic differentiation)



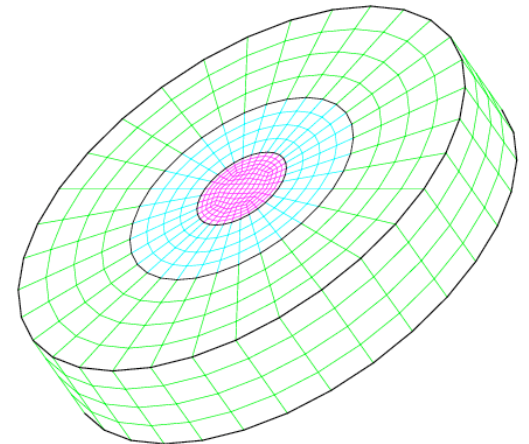
POCs: Pavel Bochev, Denis Ridzal, Kara Peterson
{pbboche,dridzal,kjpeter}@sandia.gov

PAMGEN

In-line Meshing Library

- On-the-fly parallel generation of simple meshes.
 - ◆ 2D/3D, quad/hex
 - ◆ Can mesh cylindrical, block, tubular domains
 - ◆ No interprocess communication required
 - ◆ Fixed mesh-size per process easily generated for weak scaling experiments.
 - ◆ No pre-partitioned mesh files needed.
 - ◆ C interface to local mesh geometry and topology as well as inter-processor connections.
- New in Trilinos 10:
 - ◆ Extended to $> 2B$ elements (ordinal type is user-specified).

POC: Dave Hensinger dmhensi@sandia.gov





Whirlwind Tour of Packages

Frequently Requested Discretizations **Methods** Core Solvers/
Preconditioners



Sacado: Automatic Differentiation

- Efficient OO based AD tools optimized for element-level computations
- Applies AD at “element”-level computation
 - ♦ “Element” means finite element, finite volume, network device,...
- Template application’s element-computation code
 - ♦ Developers only need to maintain one templated code base
- Provides three forms of AD
 - ♦ Forward Mode: $(x, V) \longrightarrow \left(f, \frac{\partial f}{\partial x} V\right)$
 - Propagate derivatives of intermediate variables w.r.t. independent variables forward
 - Directional derivatives, tangent vectors, square Jacobians, $\partial f / \partial x$ when $m \geq n$.
 - ♦ Reverse Mode: $(x, W) \longrightarrow \left(f, W^T \frac{\partial f}{\partial x}\right)$
 - Propagate derivatives of dependent variables w.r.t. intermediate variables backwards
 - Gradients, Jacobian-transpose products (adjoints), $\partial f / \partial x$ when $n > m$.
 - ♦ Taylor polynomial mode: $x(t) = \sum_{k=0}^d x_k t^k \longrightarrow \sum_{k=0}^d f_k t^k = f(x(t)) + O(t^{d+1}), f_k = \frac{1}{k!} \frac{d^k}{dt^k} f(x(t))$
 - ♦ Basic modes combined for higher derivatives.

Developers: Eric Phipps etphipp@sandia.gov
 David Gay dmg@acm.org



Whirlwind Tour of Packages

Frequently Requested Discretizations Methods **Core** Solvers/
Preconditioners



Thyra

- High-performance, abstract interfaces for linear algebra
- Offers flexibility through abstractions to algorithm developers
- Linear solvers (Direct, Iterative, Preconditioners)
 - ♦ Abstraction of basic vector/matrix operations (dot, axpy, mv).
 - ♦ Can use any concrete linear algebra library (Epetra, PETSc, BLAS).
- Nonlinear solvers (Newton, etc.)
 - ♦ Abstraction of linear solve (solve $Ax=b$).
 - ♦ Can use any concrete linear solver library:
 - AztecOO, Belos, ML, PETSc, LAPACK
- Transient/DAE solvers (implicit)
 - ♦ Abstraction of nonlinear solve.
 - ♦ ... and so on.

POC: Roscoe Bartlett rbbartl@sandia.gov

Stratimikos

- Defines class `Thyra::DefaultLinearSolverBuilder`.
- Provides common access to:
 - Linear Solvers: `Amesos`, `AztecOO`, `Belos`, ...
 - Preconditioners: `Ifpack`, `ML`, ...
- Reads in options through a `parameter list` (read from XML?)
- Accepts any linear system objects that provide
 - `Epetra_Operator` / `Epetra_RowMatrix` view of the matrix
 - SPMD vector views for the RHS and LHS (e.g. `Epetra_[Multi]Vector` objects)
- Provides `uniform access` to linear solver options that can be leveraged `across multiple applications and algorithms`

POC: Roscoe Bartlett rbbartl@sandia.gov



Stratimikos Parameter List and Sublists

```
<ParameterList name="Stratimikos">
  <Parameter name="Linear Solver Type" type="string" value="Aztec00"/>
  <Parameter name="Preconditioner Type" type="string" value="Ifpack"/>
  <ParameterList name="Linear Solver Types">
    <ParameterList name="Amesos">
      <Parameter name="Solver Type" type="string" value="Klu"/>
      <ParameterList name="Amesos Settings">
        <Parameter name="MatrixProperty" type="string" value="general"/>
        ...
      <ParameterList name="Mumps"> ... </ParameterList>
      <ParameterList name="Superludist"> ... </ParameterList>
    </ParameterList>
  </ParameterList>
  <ParameterList name="Aztec00">
    <ParameterList name="Forward Solve">
      <Parameter name="Max Iterations" type="int" value="400"/>
      <Parameter name="Tolerance" type="double" value="1e-06"/>
      <ParameterList name="Aztec00 Settings">
        <Parameter name="Aztec Solver" type="string" value="GMRES"/>
        ...
      </ParameterList>
    </ParameterList>
    ...
  </ParameterList>
  <ParameterList name="Belos"> ... </ParameterList>
</ParameterList>
<ParameterList name="Preconditioner Types">
  <ParameterList name="Ifpack">
    <Parameter name="Prec Type" type="string" value="ILU"/>
    <Parameter name="Overlap" type="int" value="0"/>
    <ParameterList name="Ifpack Settings">
      <Parameter name="fact: level-of-fill" type="int" value="0"/>
      ...
    </ParameterList>
  </ParameterList>
  <ParameterList name="ML"> ... </ParameterList>
</ParameterList>
</ParameterList>
```

Top level parameters

Linear Solvers

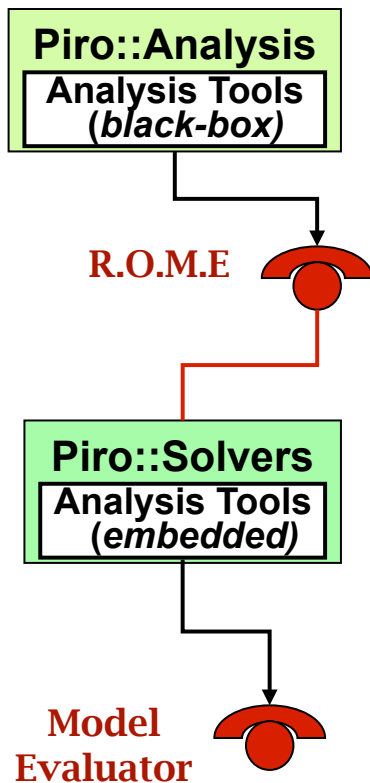
**Sublists passed
on to package
code!**

Preconditioners

**Every parameter
and sublist is
handled by Thyra
code and is fully
validated!**



Piro: Single wrapper for all solver and analysis packages



Piro::Analysis abstraction:

- Wraps Dakota, MOOCHO, OptiPack analysis packages

Constructed with: Solver, paramList
Will make use of analytic sensitivities

Piro::Solver abstraction:

- Wraps NOX, LOCA, Rythmos, Stokhos, MOOCHO solvers

Constructed with: modelEvaluator, paramList
Calculates sensitivities
Can be used to construct Piro::Analysis layer

With Piro, you can select all these through a parameter list:

- Optimization of transient problem
- UQ of a steady problem
- Least-squares fit of bifurcation problem

POC: Andy Salinger
agsalin@sandia.gov



“Skins”

- PyTrilinos provides Python access to Trilinos packages
- Uses SWIG to generate bindings.
- Epetra, AztecOO, IFPACK, ML, NOX, LOCA, Amesos and NewPackage are supported.

POC: Bill Spatz spatz@sandia.gov

- WebTrilinos: Web interface to Trilinos
- Generate test problems or read from file.
- Generate C++ or Python code fragments and click-run.
- Hand modify code fragments and re-run.
- **Will use during hands-on.**

POC: Jim Willenbring jmwille@sandia.gov



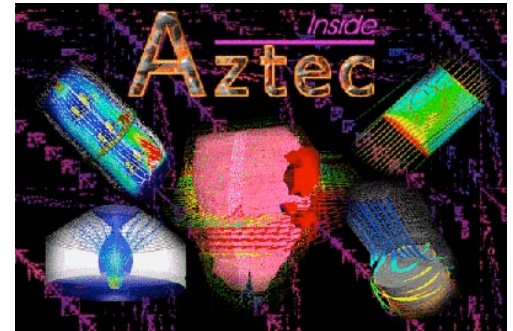
Whirlwind Tour of Packages

Frequently Requested Discretizations Methods Core **Solvers/
Preconditioners**



AztecOO

- Krylov subspace solvers: CG, GMRES, Bi-CGSTAB,...
- Incomplete factorization preconditioners
- Aztec is the workhorse solver at Sandia:
 - ♦ Extracted from the MPSalsa reacting flow code.
 - ♦ Installed in dozens of Sandia apps.
 - ♦ 1900+ external licenses.
- AztecOO improves on Aztec by:
 - ♦ Using Epetra objects for defining matrix and RHS.
 - ♦ Providing more preconditioners/scalings.
 - ♦ Using C++ class design to enable more sophisticated use.
- AztecOO interfaces allows:
 - ♦ Continued use of Aztec for functionality.
 - ♦ Introduction of new solver capabilities outside of Aztec.



POCs: Mike Heroux, Alan Williams, Ray Tuminaro
{maherou,william,rstumin}@sandia.gov

Teko

(not released yet)

- Facilitates implementing “block” preconditioners
- Target applications are multiphysics systems
- Implemented preconditioners
 - ♦ Block 2x2 LU Multiplicative
 - ♦ Block Gauss-Seidel LSC (Navier-Stokes)
 - ♦ Block Jacobi SIMPLE (Navier-Stokes)
 - ♦ Additive
- Can use Trilinos solvers/preconditioners for sub solves
- Can convert monolithic system into blocked system
- Can be used by Aztec and Belos
- Can use ML, Ifpack

POC: Eric Cyr (eccyr@sandia.gov)





Full Vertical Solver Coverage



| | | | |
|--|---|--|---|
| Optimization Unconstrained: Constrained: | Find $u \in \mathbb{R}^n$ that minimizes $g(u)$ Find $x \in \mathbb{R}^m$ and $u \in \mathbb{R}^n$ that minimizes $g(x, u)$ s.t. $f(x, u) = 0$ | Sensitivities (Automatic Differentiation: Sacado) | MOOCHO |
| Bifurcation Analysis | Given nonlinear operator $F(x, u) \in \mathbb{R}^{n+m}$ For $F(x, u) = 0$ find space $u \in U \ni \frac{\partial F}{\partial x}$ | | LOCA |
| Transient Problems DAEs/ODEs: | Solve $f(\dot{x}(t), x(t), t) = 0$ $t \in [0, T], x(0) = x_0, \dot{x}(0) = x'_0$ for $x(t) \in \mathbb{R}^n, t \in [0, T]$ | | Rythmos |
| Nonlinear Problems | Given nonlinear operator $F(x) \in \mathbb{R}^m \rightarrow \mathbb{R}^m$ Solve $F(x) = 0 \quad x \in \mathbb{R}^n$ | | NOX |
| Linear Problems Linear Equations: Eigen Problems: | Given Linear Ops (Matrices) $A, B \in \mathbb{R}^{m \times n}$ Solve $Ax = b$ for $x \in \mathbb{R}^n$ Solve $A\nu = \lambda B\nu$ for (all) $\nu \in \mathbb{R}^n, \lambda \in \mathbb{R}$ | | AztecOO Belos Ifpack, ML, etc... Anasazi |
| Distributed Linear Algebra Matrix/Graph Equations: Vector Problems: | Compute $y = Ax; A = A(G); A \in \mathbb{R}^{m \times n}, G \in \mathfrak{S}^{m \times n}$ Compute $y = \alpha x + \beta w; \alpha = \langle x, y \rangle; x, y \in \mathbb{R}^n$ | | Epetra Tpetra |



Trilinos / PETSc Interoperability

- Epetra_PETScAIJMatrix class
 - ♦ Derives from Epetra_RowMatrix
 - ♦ Wrapper for serial/parallel PETSc aij matrices
 - ♦ Utilizes callbacks for matrix-vector product, getrow
 - ♦ No deep copies
- Enables PETSc application to construct and call virtually any Trilinos preconditioner
 - ♦ ML, Ifpack, AztecOO, ...
 - ♦ All Trilinos options immediately available via parameter lists
- ML accepts fully constructed PETSc KSP solvers as smoothers
 - ♦ Fine grid only
 - ♦ Assumes fine grid matrix is really PETSc aij matrix
 - ♦ Complements Epetra_PETScAIJMatrix class
 - For any smoother with getrow kernel, PETSc implementation should be *much* faster than Trilinos
 - For any smoother with matrix-vector product kernel, PETSc and Trilinos implementations should be comparable



Trilinos Integration into an Application

Where to start?

<http://trilinos.sandia.gov>

Obtaining & Building Trilinos

- Source available from trilinos.sandia.gov/download
- Configuration requires cmake, version 2.8 (www.cmake.org)
- [Example](#): enable ML package, ML examples, all optional packages:

serial

```
cmake \  
-D CMAKE_INSTALL_PREFIX:PATH="${PWD}" \  
-D Trilinos_ENABLE_ML:BOOL=ON \  
-D Trilinos_ENABLE_EXAMPLES:BOOL=ON \  
/home/jhu/trilinos-10.4.0-Source
```

parallel

```
cmake \  
-D CMAKE_INSTALL_PREFIX:PATH="${PWD}" \  
-D Trilinos_ENABLE_ML:BOOL=ON \  
-D Trilinos_ENABLE_EXAMPLES:BOOL=ON \  
-D TPL_ENABLE_MPI:BOOL=ON \  
-D Trilinos_EXTRA_LINK_FLAGS:STRING="-lgfortran" \  
/home/jhu/trilinos-10.4.0-Source
```

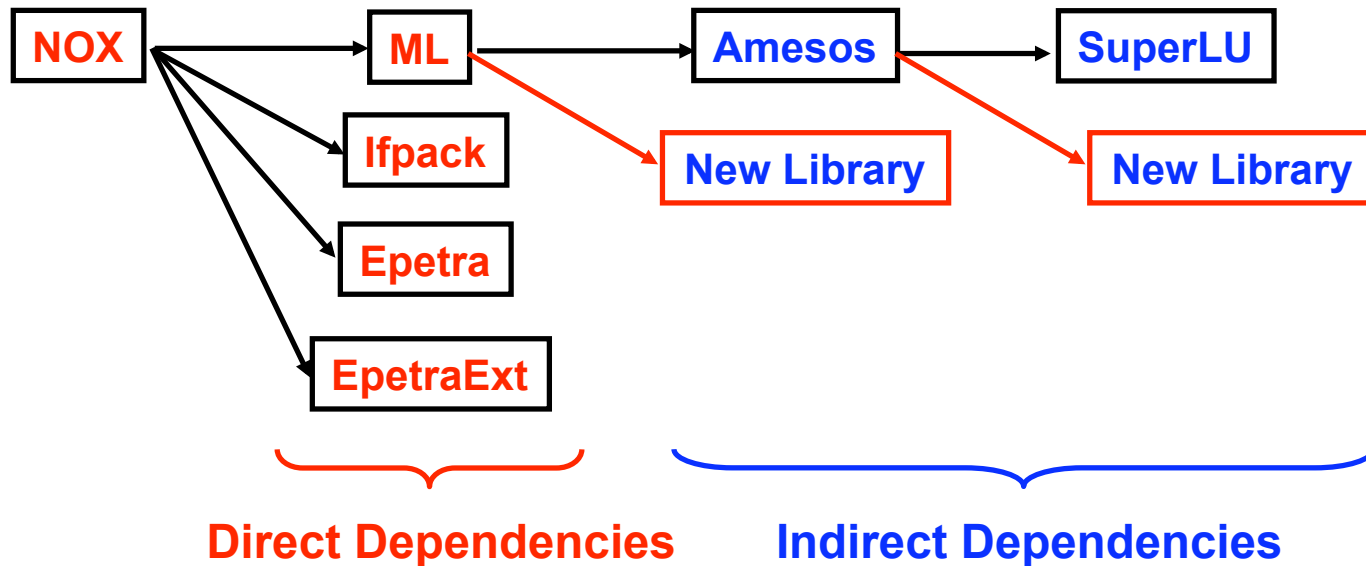
- cmake fine-grain control
 - ♦ User can configure just those packages she wants
 - ♦ Can activate tests and/or examples on per-package basis.
- Many, many more cmake options
- [More help](#):
 - ♦ trilinos.sandia.gov/Trilinos10CMakeQuickstart.txt
 - ♦ Many configure scripts in trilinos-10.4.0-Source/sampleScripts

Linking Your Application with Trilinos

- Once Trilinos is built, how do you link against the application?
- There are a number of issues:
- Library link order:
`-lnoxepetra -lnox -lepetra -lteuchos -lblas -llapack`
- Consistent compilers:
`g++, mpiCC, icc, ...`
- Consistent build options and package defines:
`g++ -g -O3 -D HAVE_MPI -D _STL_CHECKED`
- Answer: Export Makefile system

Why Export Makefiles are Important

- Release 10.4 has 47 packages.
- As new package dependencies (especially optional ones) are introduced, more maintenance is required by top-level packages:



A top level package like NOX must either:

- Account for the new libraries in its configure script (not scalable)
- Depend on direct dependency packages to supply them through “export” Makefiles.

Export Makefile in Action

```
#####
## Example Makefile that builds "NOX_1DFEMNonlinear" example outside of Trilinos
#####
## Set the Trilinos install directory
TRILINOS_INSTALL_DIR = /home/jhu/Trilinos/trilinos-release-10-4-branch/sandbox-nox-mpi

## Include any direct Trilinos library dependencies - in this case only nox
include $(TRILINOS_INSTALL_DIR)/include/Makefile.export.NOX

COMPILE_FLAGS = $(NOX_CXX_FLAGS) $(NOX_INCLUDE_DIRS)
# TPL => "Third Party Library"
LINK_FLAGS = $(NOX_LIBRARY_DIRS) $(NOX_LIBRARIES) $(NOX_TPL_LIBRARY_DIRS) $(NOX_TPL_LIBRARIES)
              $(NOX_EXTRA_LD_FLAGS)

## ## Build your application code ##
NOX_epetra_1DFEMNonlinear_Example.exe: Example.o Basis.o Problem_Interface.o FiniteElementProblem.o
    $(NOX_CXX_COMPILER) $(NOX_CXX_FLAGS) -o NOX_epetra_1DFEMNonlinear_Example.exe Example.o Basis.o
    Problem_Interface.o FiniteElementProblem.o $(LINK_FLAGS)

Basis.o: Basis.C
    $(NOX_CXX_COMPILER) $(COMPILE_FLAGS) -c Basis.C
Example.o: Example.C
    $(NOX_CXX_COMPILER) $(COMPILE_FLAGS) -c Example.C
FiniteElementProblem.o: FiniteElementProblem.C
    $(NOX_CXX_COMPILER) $(COMPILE_FLAGS) -c FiniteElementProblem.C
Problem_Interface.o: Problem_Interface.C
    $(NOX_CXX_COMPILER) $(COMPILE_FLAGS) -c Problem_Interface.C

clean:
    \rm -f *.o *.exe *~
```



Concluding Remarks



Trilinos Availability / Information

- Trilinos and related packages are available via LGPL.
 - ◆ Current release (10.4) is “click release”. Unlimited availability.
 - ◆ Next release: September 2010.
- Trilinos Awards:
 - ◆ 2004 R&D 100 Award.
 - ◆ SC2004 HPC Software Challenge Award.
 - ◆ Sandia Team Employee Recognition Award.
 - ◆ Lockheed-Martin Nova Award Nominee.
- More information:
 - ◆ <http://trilinos.sandia.gov>
- 7th Annual Trilinos User Group Meeting was November 2009 @ SNL
 - ◆ talks available for download
- Next TUG is November 2010 at Sandia/Albuquerque. Contact Jim Willenbring (jmwille@sandia.gov) if you are interested in attending.

What's new in Trilinos 10.4

- July/2010 release includes 47 packages
- 1st release packages
 - ◆ Ctrilinos – provides C bindings for Trilinos C++ packages. Target audience is Fortran application developers.
 - ◆ Globipack – collection of scalar-only globalization code to support line searches
 - ◆ Optipack – Thyra-based ANA optimization algorithms, e.g., nonlinear CG algorithms
- Makefile export system under cmake
 - ◆ Makes linking against Trilinos easier

Useful Links

Trilinos website: trilinos.sandia.gov

Trilinos tutorial: code.google.com/p/trilinos/wiki/TrilinosHandsOnTutorial

Trilinos mailing lists: trilinos.sandia.gov/mail_lists.html

Trilinos User Group (TUG) meetings:

trilinos.sandia.gov/events/trilinos_user_group_2009

trilinos.sandia.gov/events/trilinos_user_group_2008